



Bridgewater State University

Virtual Commons - Bridgewater State University

Honors Program Theses and Projects

Undergraduate Honors Program

4-30-2019

The Education of Digital Design for Children

Neha Kharidi

Bridgewater State University

Follow this and additional works at: https://vc.bridgew.edu/honors_proj



Part of the [Computer Sciences Commons](#)

Recommended Citation

Kharidi, Neha. (2019). The Education of Digital Design for Children. In *BSU Honors Program Theses and Projects*. Item 381. Available at: https://vc.bridgew.edu/honors_proj/381
Copyright © 2019 Neha Kharidi

This item is available as part of Virtual Commons, the open-access institutional repository of Bridgewater State University, Bridgewater, Massachusetts.

The Education of Digital Design for Children

Neha Kharidi

Submitted in Partial Completion of the Requirements for
Commonwealth Honors in Computer Science

Bridgewater State University

April 30, 2019

Dr. Michael Black, Thesis Advisor
Dr. Heather Pacheco Guffrey, Committee Member
Dr. Haleh Khojasteh, Committee Member

Contents

Abstract	3
Introduction	4
Purpose	4
Background	5
Review of Existing Simulators	6
Detailed Description of Proposed App	8
General Structure.....	8
Main Menu.....	9
Lesson Videos.....	9
Truth Tables.....	10
Simulator.....	10
Tutorial Button.....	11
Controlling Inputs.....	12
Connecting Busses.....	13
Wrong Connections.....	14
Future Work	14
Discussion	15
Conclusion	15
References	16

Abstract

Digital design is a set of rules stating how computer software and hardware are joined together and interact to make a computer work. This is an important area of study that should be made more accessible to children, particularly of the middle school age. Although middle schools do provide access to computer science software, access to the hardware aspect is nonexistent. In this project, I proposed to create a mock up for an android app that aims to teach children beginner concepts in digital design. It is my goal to one day create an app in order to bring these concepts into middle schools.

Introduction

The use of technology is rapidly increasing in today's world, and the curriculum in our middle schools no longer reflects that. Children should have more access to fields beyond basic subjects such as reading and math. However, if a child did have interest beyond just reading and math, he or she could go online and learn how to code and program in any language they want. Some schools even have computer science programming classes. In fact, in states like Alabama, funds totaling up to 675k have been allocated for the Middle School Programming Initiative in 2018 (CSTA, 2018). However, although these programs educate on the software aspect, they fail to educate on the hardware aspect, specifically they fail to educate on digital design. Digital design is concerned with the way hardware components are connected together to form a computer system. There is a large population of people who can use a computer but have no resource to learn how one actually works or thinks. We are living in a fast-paced world that is driven by electronics, and the use of technology in our lives increases more and more every day. However, the problem is that there are not enough people interested in the field of computer science hardware due to the lack of exposure. In a TED talk published in 2015, Ashley Gavin, a computer science education consultant, talked about how there are more than 1 million jobs going unfilled because by the time kids go into college and pick their major, more than half of them have no idea what computer science is (Gavin, 2015). As the technology in this world is changing, the need to supply people with the tools to keep up with the rapidly changing technology remains essential. It is important to educate the upcoming generation.

It is no longer just engineers who benefit from learning digital design. There are many areas of study that overlap with not only computer science but the hardware of computer science. For example, physics and computer hardware have a lot of overlap.

Furthermore, if someone wanted to further their knowledge in the software aspect of computer science there are many online resources available dedicated to this cause. However, there is a severe lacking in resources that would educate on the hardware aspect. This project aims to take the first steps toward creating a tool towards creating a tool that solves this problem; it will help middle school age children learn the structure and behavior of the various components of a computer, and how they interact to process the needs of the user.

Purpose

My goal is to create a mock up for an application that serves as a tool to help children at the

middle school age become familiar with the basics of the hardware components in a computer and the fundamentals of logic design, so that they may gain exposure to the field early. This provides them with the avenue to possibly elucidate a career in computer science hardware early on.

Background

There are almost no resources easily available to teach computer science hardware, and the ones that do exist are too complicated for a novice learner to use. Some of these resources include WebMIPS, ProcSim and SimuProc. Each of these tools works by loading in code, which is not in the scope of the app that I am trying to design.

Before I began creating these mock ups, it was necessary to understand the motivation behind children's learning. In order to do this, I spent some time working with education professor Heather Pacheco-Guffrey to understand the motivations behind learning in middle school aged children. Relevant studies have indicated that digital game-based learning (DGBL) possesses significant potential for increasing students' learning motivation (Huang, 2011; Papastergiou, 2009) and enhancing their learning effectiveness (Yang, 2012; Wang & Chen, 2010). I learned that learning by doing is a core constructive principle when it comes to learning new concepts. With this new knowledge, I proposed to learn the basics of android app development in order to create detailed mockups of an app that teaches computer hardware particularly related to circuits. I intended to include various topics pertaining to digital components used in the organization and design of digital computers. These topics would include logic gates, truth tables and boolean algebra.

In order to understand how an Android app works from the inside and create realistic mock ups, I took online classes from Udacity.com. In these courses, I reviewed layouts and how to position items on a page relative to each other. This helped me understand what kind of functions an app should include, so I could make realistic mock ups that can be replicated when made into a functioning app.

Finally, I researched a number of already available tools that deal with digital design. This gave me an idea of how concepts in this field are already being depicted to an audience. I tested each of these tools and made a note of various aspects that would be applicable to my app. When I tested these tools, I mainly inspected the usability of three main concepts; logic gates, truth tables and boolean algebra. I chose these topics because they make up the foundation of digital

design in accordance with the *Computer System Architecture third edition* textbook. When I tested these tools, I noted how complicated it was to use these features; specifically, I noted the level of difficulty in dragging logic gates, recognizing and defining gates and snap grid systems and ease in drawing wires

Review of Existing Simulators

While researching existing tools for learning digital design, I found 3 notable ones: Visual Paradigm online, Circuitverse and Logic.ly. When reviewing these tools I focused on the way the logic gates were displayed, whether the interface I'm using had a snap grid system, and how easy it was to connect wires. I focused on these topics because they connect to the biggest concept in digital design: drawing circuits. In the table below, there are three simulators listed. Each of these simulators are listed with their name and qualities they possess pertaining to the four ideas mentioned above.

Each of these tools were chosen because their graphical interfaces possessed features that made using the simulator easy to grasp. These simulators are not intended for new learning. In order to incorporate an educational perspective, I used *Computer System Architecture* in conjunction with design elements of the simulators.

Table 1. Notable Existing Simulators

Name	1)Ability to easily drag logic gates 2)Clarity of gates/label/input output legs/ 3)Snap grid system 4)Quality of wires
Visual Paradigm online	1)Dragging gates is simple 2)Thin black lines, when mouse hovers over gate the label is shown, input/output legs are thin black lines 3)Uses grid system 4)Wires are lines, easy to connect with sharp lines
Circuitverse	1)Dragging gates is simple 2) Thick black lines, when mouse hovers over gate the label is shown, input/output legs are

	green circles 3)Does not use a grid system 4)Wires are lines that connect with sharp lines, easy to connect, red color
Logic.ly	1)Dragging gates is simple 2) Thick black lines, label is listed under each gate, input/output legs are lines with a circle attached. 3)Does not use a grid system 4)Wires are thick red lines, with no sharp lines

Each of these tools also had some design features that have been incorporated into the mock ups.

Image 1, Visual Paradigm AND Gate

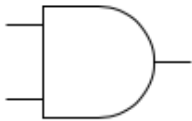
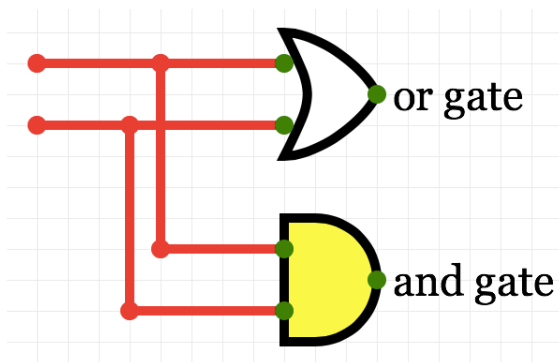


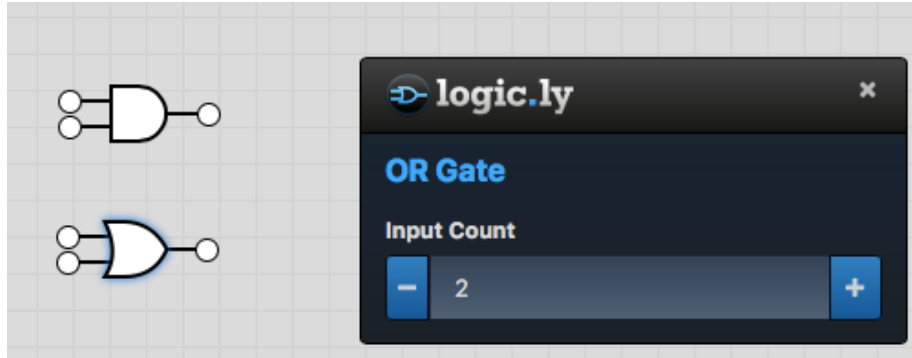
Image 1 is an example of an AND gate used in Visual Paradigm. This website uses simpler logic gates with clear input and output legs. This shape is similar to the shapes found in textbooks. Computer System Architecture in particular uses similar symbols for gates.

Image 2, Circuitverse



Circuitverse includes a feature that allows for adding a label.

Image 3, Logic.ly



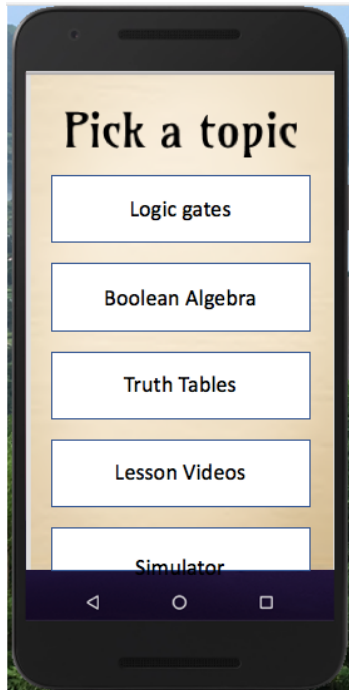
Logic.ly includes a window that allows the user to control each gate and its inputs.

Detailed Description of Proposed App

General Structure

My proposed app is an Android application that will teach the basics of computer architecture to middle school age children. The app starts on a menu page where the student can pick topics including logic gates, Boolean algebra, truth tables, or simulator. Each of these leads to their respective activity page. There is also a choice for lesson videos, where the student can be led to chapter videos. These videos will give example problems and real-world connections and activities for the student to try in the simulator.

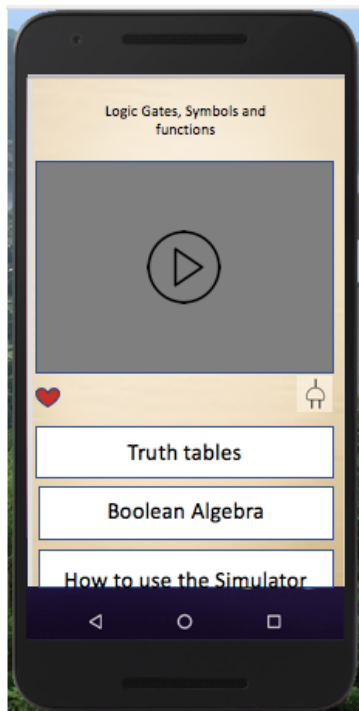
Image 4, Main Menu



The first page is the main menu. Selecting a choice leads to its respective activity. “Logic gates” leads to a static page of logic gates, with their name and symbol. “Boolean algebra” leads to a static page of Boolean algebra laws, with symbols and rules. “Truth tables” lead to the automatic table generator. “Lesson videos” leads to all the instructional videos and activities, and the student can follow those videos by switching to the simulator.

The topics have been listed in the order that they are taught in the Intro to Computer Organization class taught at Bridgewater State University.

Image 5, Lesson Videos

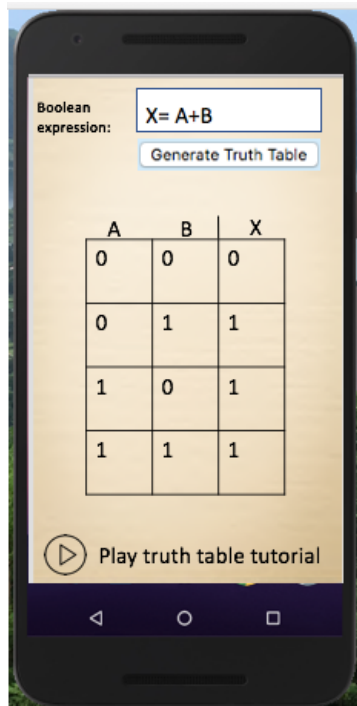


The lesson videos section is a place in the app meant for learning new information. The student can choose the topic they want and follow along in the simulator. The little logic gate under the video box is a quick link to the simulator, so that the student can follow along while watching the videos.

The student can also refer back the “How to use the simulator” section, so if there is any confusion with usage, the video can answer those questions.

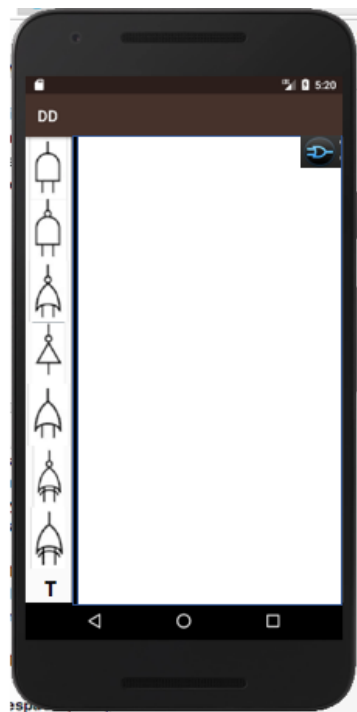
If the student likes a certain video or wants to keep a list of favorite videos, they can click on the heart button which will allow the user to save the video in a separate list.

Image 6, Truth Table



In my proposed mock ups, I have included a truth table section. The user can input an equation and obtain an auto generated truth table with the various outputs. If the student wants to be reminded on how the outputs are calculated, they can click on the link on the bottom of the screen. This feature is important because as the equations get more complicated, the various input/output possibilities grow. Therefor having a feature that allows the user to get all the possibilities at once eliminates some confusion.

Image 7 Simulator



The simulator is the main feature of the app. It's the most important aspect in learning digital design. The user can draw different circuits using the logic gates on the left-hand side. At the bottom of the list of logic gates, I have included a tutorial button. When the button is pressed, a tutorial pops up (Image 8). In the top right corner, the user can control input values of each gate that is placed on screen (Image 10).

In my attempt to create these mock ups, I learned basic app design in order to create realistic mock ups. I learned how to set layouts; for example, the following screen would be made by setting an overall horizontal layout consisting of all the gates and the white blank screen. Followed by an inner vertical layout, consisting of all the logic gates.

Image 8 & 9, Tutorial Button

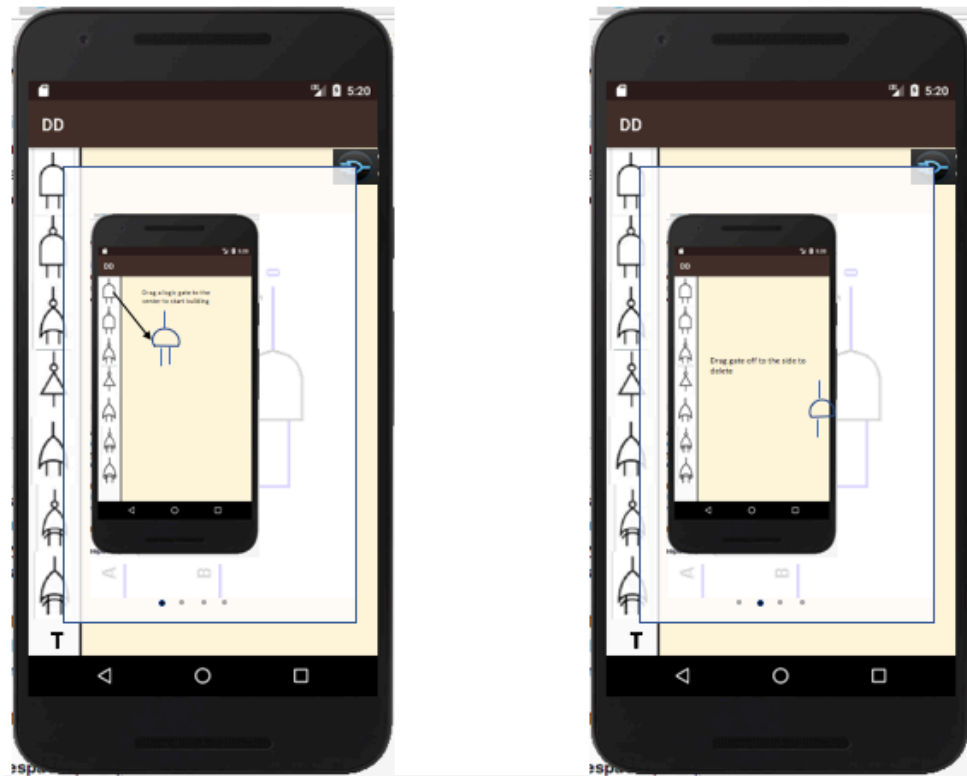
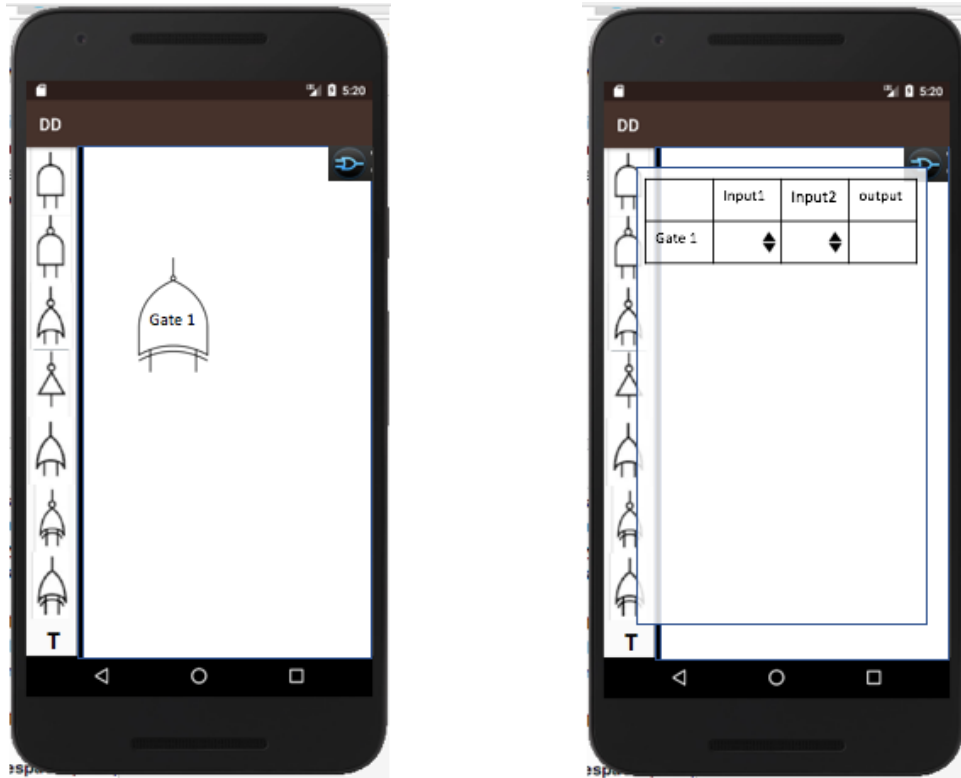


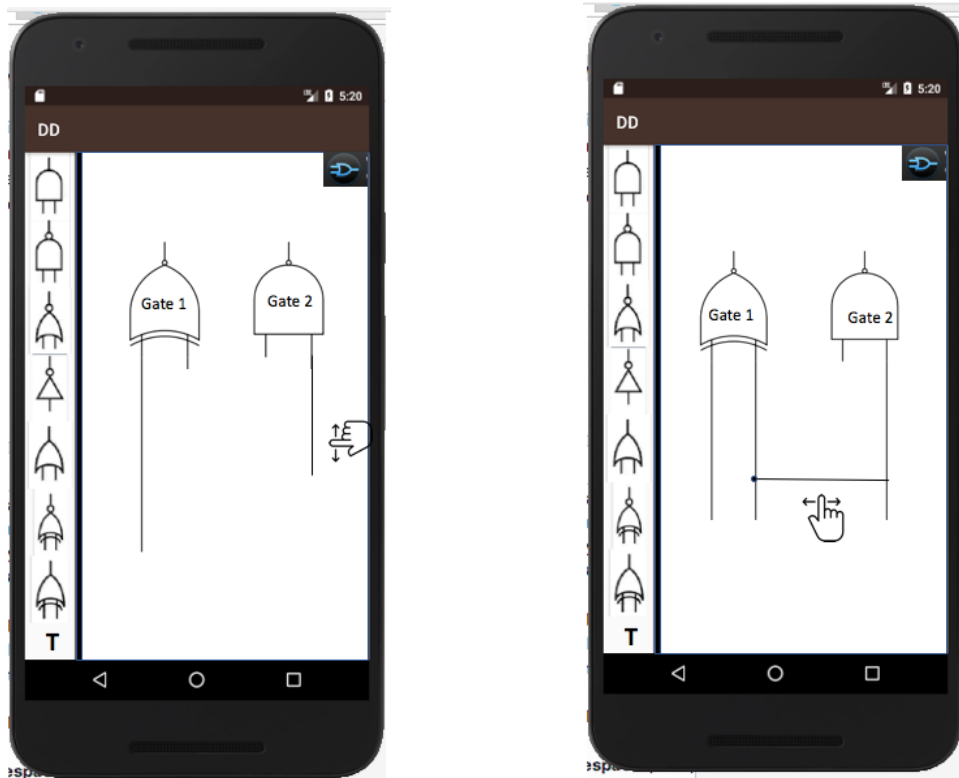
Image 8 is an example of how the tutorial section would work. This section is different from the “How to use the simulator” option available in the lesson videos section. This version of tutorials would include important tips and tricks the user can flip through to remind themselves of different functions. I chose to have the user flip through the help screens because the action of sliding through the screens helps keep the attention of the user as opposed to having the reader lose interest, reading through blocks of text.

Image 10 & 11, Controlling Inputs



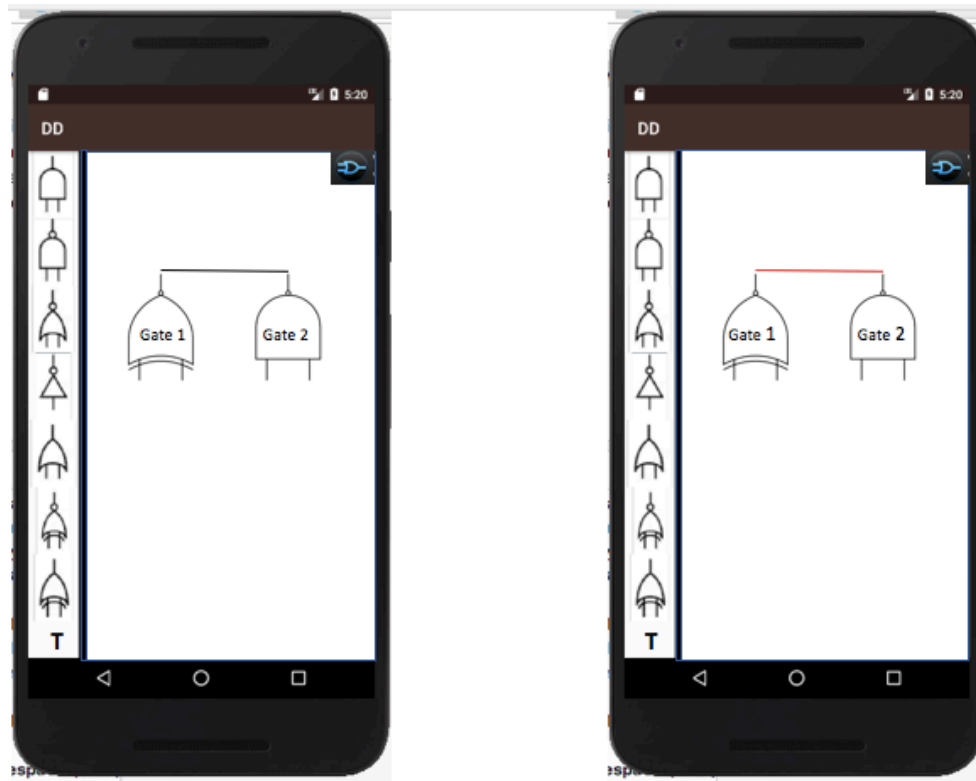
Inspired by Circuitverse I proposed this app would auto generate a name when a user drags a gate to the screen. This feature avoids confusion by helping the user distinguish between the different gates. Users can then automatically control the inputs by clicking the top right button; this feature was inspired by Logic.ly. I chose to incorporate this idea because it gives a sort of separation from the circuits and the control of inputs for the gates. Additionally, if the user wants to get rid of a gate, they will be able to drag it off to the side and it will disappear off the screen.

Image 12 & 13, Connecting Busses



The user can connect the gates with each other by dragging from the leg of one gate and touching the leg of another gate. This layout was the simplest to keep track of, which was apparent in my use of Visual Paradigm. Keeping the look of the busses simple is necessary when working with such complicated concepts. If the screen becomes too cluttered then the user is likely to become demotivated. I also have the user drag the gates on and off the screen, to give the user the illusion that they are in control. According to an article written by the University of Graz, “The challenge of the game should fit the ability of the player; otherwise he/she would experience either anxiety or boredom instead of a ‘flow’ . By this means the player has the feeling of controlling the situation,” (Mattheiss 2009; Albert 2009). This is exactly what I sought out to do when creating my designs; I want the user to be in control of the situation and not feel a sense of panic when the concepts get harder. The user can delete these busses by dragging them to the edge same as the logic gates, essentially leaving all the design choices up to the user.

Image 14 & 15, Wrong Connections



When an incorrect connection is made, the user is notified by the bus turning red. Red is commonly associated with wrong actions, so when the wire turns red, the user will identify their action is incorrect.

Future Work

While the mock ups I have designed are a good start, there is much more to be done. Down the road, I would like to add flip flop circuits and activities for the user to try and solve. Ideally, the app would cover a lot more concepts pertaining to digital design. I would also like to look deeply into user interface design to make the styles of the final app more appealing to look at. When it comes to learning in digital environments, the incorporation of game behavior is important. This is a quality I have not yet been able to plan into my designs, but it is an important one. I would like to add game qualities in the simulator, such as having the user draw a specific circuit in the simulator based off an equation. The users would learn how to do this with the instruction videos and the practice they would ideally acquire in the simulator.

I would also like to start developing the app, and hopefully introduce it into classrooms. I would like to do a user study to collect statistics on the amount of time a beginner user must spend in

order to get a grasp on the subject matter. Once those statistics are collected, I would like to refine the app to be successful to streamline the struggle a student goes through when there is a learning curve present. Once I have reached a place I am comfortable with in the app, I hope to add levels to try to reach all different levels of learners.

Discussion

When I started this project, I hoped to code the app myself. Then, I realized after some trial and error, that the scope of the project is too much considering my level of knowledge in app design. Once I realized this, I decided that the amount of app design knowledge I did have would better serve me to design a plan for the app I would have liked to create. Taking my previous research in digital game based learning and its motivational influence in middle school children, I was able to come up with a basic design for an app to be developed later.

Conclusion

The need for tools to help younger audiences learn concepts in digital design remains important. I hope that this project is a step forward in solving this problem.

References

“Digital Circuit Simulator Online.” CircuitVerse, circuitverse.org/simulator.

E. Mattheiss, M. D. Kickmeier-Rust, C. M. Steiner & D. Albert (2009). Motivation in game-based learning: It's more than 'flow'. In Schwill, A., & Apostolopoulos, N. (Eds.), Lernen im Digitalen Zeitalter - Workshop-Band Dokumentation der Pre-Conference zur DeLFI2009 - Die 7. E-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (pp. 77-84).

Papastergiou, M. “Digital Game-Based Learning in Middle School Computer science Education: Impact on Educational Effectiveness and Student Motivation.” Egyptian Journal of Medical Human Genetics, Elsevier, 25 July 2008, www.sciencedirect.com/science/article/pii/S0360131508000845.

“Teach Logic Gates + Digital Circuits Effectively - with Logicly.” Logicly - A Logic Circuit Simulator for Windows and MacOS - Logic Gates, Flip-Flops, Computer Architecture, Electronics, Integrated Circuits, logic.ly/.

TEDxTalks. “Computer science education: why does it suck so much and what if it didn't? | Ashley Gavin | TEDxNYU.” YouTube, YouTube, 21 July 2015, www.youtube.com/watch?v=5jmN_tBS0t4.

Udacity, classroom.udacity.com/courses/ud834.

VP Online - Online Drawing Tool, online.visual-paradigm.com/diagrams.jsp#diagram:proj=0&type=LogicDiagram&gallery=/repository/f2567325-738b-4394-bdc4-c9f117e31d9d.xml&name=Alarm%20Circuit%20Transformation.

Woo, Jeng-Chung. “Digital Game-Based Learning Supports Student Motivation, Cognitive Success, and Performance Outcomes.” Department of Arts and Plastic Design, National Taipei University of Education, Taiwan, Educational Technology & Society, 2013.

2018 State of Computer science Education. (2018). Retrieved from

<https://advocacy.code.org/>